

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年   3 月 2 7 日  
Date of Application:

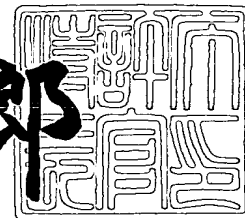
出 願 番 号            特 願 2 0 0 3 - 0 8 6 9 0 6  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 3 - 0 8 6 9 0 6 ]

出 願 人            株式会社日立製作所  
Applicant(s):

2 0 0 3 年   7 月 1 0 日

特許庁長官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号   出証特 2 0 0 3 - 3 0 5 6 2 6 0

【書類名】 特許願

【整理番号】 K03002221A

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/00

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 藺田 浩二

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 岩寄 正明

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 松並 直人

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】	要約書 1
【プルーフの要否】	要

【書類名】 明細書

【発明の名称】 計算機システム

【特許請求の範囲】

【請求項 1】

ファイル管理情報が格納された第一の記憶装置と、  
前記第一の記憶装置に接続される第一の計算機と、  
前記第一の計算機とネットワークを介して接続される第二の計算機と、  
前記第二の計算機と接続され、前記ファイル管理情報で管理されるファイルデータが格納される第二の記憶装置とを有することを特徴とする計算機システム。

【請求項 2】

前記第一の計算機は、  
前記第二の記憶装置に格納されたファイルデータを更新する要求を受信した場合、前記第一の記憶装置に格納された前記ファイル管理情報の内容を確認し、前記ファイル管理情報の内容に従って、前記第二の計算機に前記ファイルデータを更新する要求の対象となるファイルデータの更新を指示する要求を送信し、  
前記第二の計算機は、  
前記指示する要求に基づいて、前記第二の記憶装置に格納されたファイルデータを更新し、その結果を前記第一の計算機に送信することを特徴とする請求項 1 記載の計算機システム。

【請求項 3】

前記第一の計算機は、  
前記第二の記憶装置に格納されたファイルデータを読み出す要求を受信した場合、前記第一の記憶装置に格納された前記ファイル管理情報の内容を確認し、前記ファイル管理情報の内容に従って前記第二の計算機に前記ファイルデータを読み出す要求の対象となるファイルデータの読み出しを指示する要求を送信し、  
前記第二の計算機は、  
前記指示する要求に基づいて、前記第二の記憶装置に格納されたファイルデータを読み出し、読み出した前記ファイルデータを前記第一の計算機に送信し、  
前記第一の計算機は、

受信した前記ファイルデータを、前記ファイルデータを読み出す要求の発信先に送信することを特徴とする請求項 2 記載の計算機システム。

**【請求項 4】**

前記ファイル管理情報には、ファイル属性に関する情報、ファイルデータの格納位置に関する情報及びファイルへのアクセス権限に関する情報が含まれていることを特徴とする請求項 3 記載の計算機システム。

**【請求項 5】**

前記ファイル管理情報には、ある特定のファイル属性を有するファイルのみを管理するテーブルが含まれていることを特徴とする請求項 4 記載の計算機システム。

**【請求項 6】**

前記第一の計算機は、

前記ファイルデータを読み出す要求及び前記ファイルデータの更新の要求を受信した際に、これらの要求で指定されるファイルが前記テーブルに登録されているか否かを確認し、登録されている場合には、前記テーブルに対応するファイル属性に従った処理を行うことを特徴とする請求項 5 記載の計算機システム。

**【請求項 7】**

前記テーブルに対応するファイル属性が暗号化に関するファイル属性であり、  
前記第一の計算機は、

前記ファイルデータを読み出す要求を受信した際に、これらの要求で指定されるファイルが前記テーブルに登録されているか否かを確認し、登録されている場合は、前記第二の計算機から受信したファイルデータを前記テーブルに登録されている情報に基づいて暗号化して前記要求元に送信することを特徴とする請求項 6 記載の計算機システム。

**【請求項 8】**

前記ファイル管理情報には、

該計算機システムを使用する者に対する料金を計算するために必要な情報が含まれていることを特徴とする請求項 7 記載の計算機システム。

**【請求項 9】**

前記必要な情報には、前記使用する者が使用する前記第二の記憶装置に格納されているファイルデータの全容量、前記使用する者が前記第二の記憶装置へ書き込んだデータの総量及び前記使用する者が前記第二の記憶装置から読み出したデータの総量に関する情報が含まれ、これらの情報が前記第二の記憶装置を管理する前記第二の計算機ごとに管理されていることを特徴とする請求項 8 記載の計算機システム。

**【請求項 1 0】**

前記第二の計算機は、

前記第一の計算機から前記ファイルデータの更新の要求を受信した場合、前記ファイルデータの更新の要求の処理の際に前記第二の記憶装置に格納されたデータ量及びデータの更新によって増加したデータ量についての情報を前記第一の計算機に送信し、

前記第一の計算機は、受信した前記情報を用いて、前記第一の記憶装置に格納された前記必要な情報の内容を更新することを特徴とする請求項 9 記載の計算機システム。

**【請求項 1 1】**

更に第三の計算機及び前記第三の計算機に接続される第三の記憶装置を有し、  
前記第三の記憶装置は前記第二の計算機毎の課金条件についての情報を有し、  
前記第三の計算機は、前記第一の記憶装置に格納された前記必要な情報を前記第一の計算機を介して読み出し、

読み出した前記必要な情報と前記課金条件についての情報に基づいて、前記使用する者への料金を計算することを特徴とする請求項 1 0 記載の計算機システム。

**【請求項 1 2】**

前記第一の計算機及び前記第一の記憶装置は複数有り、

前記ファイル属性についての情報と前記必要な情報とは各々異なる前記第一の記憶装置に格納され、各々異なる前記第一の計算機で管理されることを特徴とする請求項 8 記載の計算機システム。

**【請求項 1 3】**

前記第二の記憶装置は複数有り、前記ファイルデータが複製されて、複数の前記第二の記憶装置の各々に格納されていることを特徴とする請求項12記載の計算機システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ネットワークを介して接続された複数の計算機を用いて実現されるシステムに関する。

【0002】

【従来の技術】

近年のインターネット等のネットワークの普及に伴い、ネットワークにおいてサービスを提供する計算機が提供するデータ量及びこれら計算機をネットワークを介して使用する使用者（以下「ユーザ」）の数が増大してきている。

【0003】

しかし、一つの計算機がユーザが要求するデータ全てを管理する従来の集中型の計算機システムでは、計算機の処理性能に限界があり、その限界性能を超えてすべてのユーザが要求するデータを供給できない。また、計算機に接続された記憶装置の記憶容量にも限界がある。さらに、その計算機に障害が発生した場合には、データへのアクセスが不可能になる。

【0004】

このような問題を解決する技術として、分散型の計算機システムが提案されている。

【0005】

例えば非特許文献1に開示されている技術では、地理的に分散した複数の場所（以下「サイト」）に設置された計算機あるいは記憶装置の各々にデータを複製して配置する。このため、複数の計算機間でデータ処理に対する負荷やデータを分散することができる。さらに、ある計算機に障害が発生した場合でも、他の計算機にあるデータの複製を用いてデータを再構成することを可能としている。

【0006】

以下、ユーザや計算機が使用するプログラム等の一かたまりのデータを「ファイル」と称する。

【 0 0 0 7 】

【非特許文献 1】

“OceanStore: An Architecture for Global-Scale Persistent Storage” ,John Kubiawicz et al., Appears in Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLoS 2000), November 2000. の 2 章。

【 0 0 0 8 】

【発明が解決しようとする課題】

従来の集中管理型の計算機システムや非特許文献 1 に開示された計算機が世界規模に分散した計算機システム環境では、多数のユーザが計算機が管理するファイルにアクセスする。このような環境でセキュリティを確保するため、計算機システムではユーザ毎にファイルへのアクセス権を設定することが必要となる。また、多数のユーザが要求する様々な用途に応じるため、このような計算機システムにおいては、個々のファイルに対し、用途に応じたファイル属性を状況に応じて設定することが必要となる。

【 0 0 0 9 】

ところが従来の計算機で実行されるファイルを管理するプログラム（以下「ファイルシステム」）では、ファイルに設定できるファイル属性の数や種類に制限があった。また、ファイル属性がファイルシステム全体で固定して管理されているので、個々のファイルに与えられたファイル属性の種類や数を自由に増減することはできず、ファイル属性の管理に柔軟性がなかった。

【 0 0 1 0 】

また、複数サイトに存在する計算機が管理する記憶装置にファイルを複製して格納するサービスをストレージサービスプロバイダ等が行う際、プロバイダとしては、各サイトの計算機が管理する記憶装置毎に異なる課金条件（以下、「課金ポリシー」）を設定し、その課金ポリシーに基づいた課金するために必要な情報（以下「課金情報」）を採取したいという要求がある。ところが、従来のファイ



ルシステムでは、ファイル属性情報として、ファイルの容量やファイルへのアクセス回数のみの固定的な情報しか採取しておらず、あるサイトでの課金ポリシーに基づいた特定の課金情報、例えばファイルを格納する記憶装置の種類等についての情報は収集できなかった。

#### 【0011】

本発明の目的は、ファイルに対して柔軟にファイル属性を追加、削除できる計算機システムを提供することである。

#### 【0012】

また本発明の別の目的は、複数のサイトの課金ポリシーを反映した課金情報を採取する計算機システムを提供することである。

#### 【0013】

##### 【課題を解決するための手段】

上記目的を達成するため、本発明の計算機システムは以下の構成とする。すなわち、計算機システムは、計算機システム内に格納されるファイルを管理するためのデータ（以下、ファイル管理情報）が格納された記憶装置を管理する第一の計算機及び第一の計算機とネットワークを介して接続され、ファイルの実データ（以下「ファイルデータ」）が格納された記憶装置を管理する第二の計算機を有する。そして、本計算機システムにおいては、ユーザからのファイルアクセス要求を受けた第一の計算機は、第一の計算機が管理するファイル管理情報をファイルアクセス要求の内容に基づいて検索し、検索された情報の内容にしたがって、ファイルアクセス要求に対応するファイルの処理要求をネットワークを介して第二の計算機に送信する。ファイルの処理要求を受信した第二の計算機は、ファイルの処理要求で指定される処理を行い、その結果を第一の計算機に送信する。結果を受信した第一の計算機は、その内容に基づいてユーザに結果を送信する。

#### 【0014】

尚、ファイル管理情報には、ファイル属性についての情報、ファイルの使用の可否についての情報及びシステムにおけるファイルの格納位置を示す情報が含まれても良い。

#### 【0015】

さらに、ファイル属性についての情報には、ある一つのファイル属性を有するファイルのみを管理するテーブルが含まれていても良い。ここで、ある一つのファイル属性としては、ファイルが暗号化されているかどうかという属性が考えられるが、これに限られない。

【0016】

また、ファイル管理情報には、課金情報、具体的には、ファイルを使用するユーザの第二の計算機毎のファイル使用に関する情報等が含まれていても良い。この場合、第一の計算機は、ファイルアクセス要求をユーザから受取った際に、そのアクセス内容に応じて第二の計算機ごとのファイル使用に関する情報を更新する。

【0017】

更に、本計算機システムは課金管理用の計算機を有し、課金管理用の計算機は、第二の計算機毎の課金情報を用いて、ファイルのユーザの料金を計算しても良い。

【0018】

尚、第一の計算機及び第二の計算機は複数でもよく、第一の計算機は、記憶装置に格納されるファイル管理情報の数だけ存在しても良い。

【0019】

更に、第二の記憶装置も複数有り、ファイルデータの複製が複数の第二の記憶装置に分散して格納されてもよい。

【0020】

【発明の実施の形態】

以下、図を用いて本発明の実施形態を説明する。

【0021】

図1は、本発明を適用した計算機システムの実施形態を示す図である。計算機システムは、サイト1500A及びサイト1500Bを有し、これらのサイト間にはインターネット1000を介して接続されている。

【0022】

サイト1500は、ユーザが使用する複数の計算機（以下「クライアント」）

800、ファイル管理情報（以下「メタデータ」）を管理する複数の計算機（以下「CN」）101、ファイルデータを管理する複数の計算機（以下「SN」）301及び複数の記憶装置600を有する。ここで、クライアント800、CN101及びSN301は、ネットワーク700を介して相互に接続されている。また、複数の記憶装置600、CN101及びSN301は、ファイバーケーブルなどで構成されたストレージネットワーク500を介して相互に接続されている。

#### 【0023】

尚、ストレージネットワーク500では、通信プロトコルとして、SCSIやファイバチャネルプロトコルが使用される。

#### 【0024】

ネットワーク700は、インターネット1000を介してサイト1500Bのネットワーク700に接続されている。したがって、サイト1の各クライアント800、CN101及びSN301は、サイト1500BのSN301Cとネットワーク700及びインターネット1000を介して通信可能である。

#### 【0025】

クライアント800、CN101及びSN301は、プロセッサ、メモリ、入出力部、ネットワークインターフェース及び記憶装置を有する計算機である。

#### 【0026】

CN101のプロセッサは、メタデータの管理を行うサーバプログラム（以下「MSVR」）100と、メタデータを格納するデータベースの管理を行うサーバプログラム（以下「DBMS」）302を実行する。これらのプログラムはメモリに格納される。一方、SN301のプロセッサは、ファイルデータの管理を行うサーバプログラム（以下「FSVR」）300を実行する。FSVR300は、SN301が有するメモリに格納される。

#### 【0027】

記憶装置600は、制御部及び記憶部を有する。記憶部に使用される記憶媒体としては、磁気ディスク、半導体ディスクまたは光ディスクがある。また、記憶装置600には、記憶部としてこれらディスクを単体で使用するディスク装置や

、記憶部として複数のディスク装置を使用するディスクアレイ等の記憶装置システムが含まれる。

#### 【 0 0 2 8 】

CN101及びSN301は、MSVR100及びFSVR300を実行して、協調して記憶装置600に格納されたファイルの管理を行う。本実施形態においては、ファイル属性や課金情報などのメタデータと、ファイルデータがそれぞれ異なる計算機及び異なる記憶装置600で管理される。これにより、より柔軟なファイル属性の管理が行える

記憶装置600Aの記憶部には、ファイル属性データベース（以下「DB」）110及び各ユーザの課金情報DB210が格納される。ファイル属性DB110には、ファイル属性テーブル120、ACLテーブル130、拡張属性テーブル140、ロケーションテーブル150及びファイルIDビットマップ160が格納される。一方、課金情報DB210には、サーバ単位課金情報テーブル220が格納される。

#### 【 0 0 2 9 】

CN101Aは、MSVR100Aを実行し、かつファイル属性DB110を用いて、記憶装置600B及びCに格納されたファイルデータに対応するファイル属性を管理する。CN101Aは、MSVR100Bを実行し、かつ課金情報DB210を用いて、計算機システムに格納されたファイルを使用する際にユーザに課せられる課金を計算するための情報を管理する。

#### 【 0 0 3 0 】

記憶装置600B及びCには、ローカルファイルシステム（以下「ローカルFS」）310が格納されている。ローカルFS310とは、ファイルデータ320、ファイルデータを管理するためのメタデータ330及びそれを管理するプログラムから構成されるものである。メタデータ330には、ローカルFS310が管理するファイルデータ320の記憶装置600における格納位置、データ量等を示す情報が格納されている。

#### 【 0 0 3 1 】

SN301は、FSVR300を実行することで、ローカルFS310を管理

制御する。尚、ローカル F S 3 1 0 A は S N 3 0 1 A に管理され、ローカル F S 3 1 0 B は S N 3 0 1 B に管理され、ローカル F S 3 1 0 C は S N 3 0 1 C に管理され、ローカル F S 3 1 0 D は S N 3 0 1 D に管理される。

#### 【 0 0 3 2 】

本実施形態の計算機システムでは、クライアント 8 0 0 がファイルを作成すると、自動的に同一内容を持つ複数のファイルが C N 1 0 1 で作成され、その複数のファイルのファイルデータは、それぞれ異なる S N 3 0 1 に転送され、それぞれが管理するローカル F S 3 1 0 に格納される。各ローカル F S 3 1 0 に格納されたファイルデータをローカルファイルと呼ぶ。

#### 【 0 0 3 3 】

本実施形態では、ひとつのファイルに対し、複数のローカルファイルが存在する。これらの各ローカルファイルの格納位置は C N 1 0 1 が決定し、格納位置を示す情報は、C N 1 0 1 の指示によって、ファイル属性 D B 1 1 0 内のロケーションテーブル 1 5 0 に登録される。また、C N 1 0 1 A は、M S V R 1 0 0 A を実行することで、ファイルサイズや作成日付などの基本的なファイル属性をファイル属性テーブル 1 2 0 で管理する。

#### 【 0 0 3 4 】

さらに、C N 1 0 1 A は、M S V R 1 0 0 A を実行することで、各クライアント 8 0 0 のファイルへのアクセス権を A C L テーブル 1 3 0 で管理する。また、C N 1 0 1 A は、M S V R 1 0 0 A を実行することで、その他のファイル属性を拡張ファイル属性テーブル 1 4 0 で管理する。

#### 【 0 0 3 5 】

一方、C N 1 0 1 B は、M S V R 1 0 0 B を実行し、かつ課金情報 D B 2 1 0 を用いて、ファイルを使用するユーザへの課金に必要な情報を管理する。

#### 【 0 0 3 6 】

一般に、S N 3 0 1 で管理される記憶装置 6 0 0 の信頼性や性能などの特性は異なるため、ユーザに対する課金ポリシーも S N 3 0 1 や記憶装置 6 0 0 毎に異なる。このため、C N 1 0 1 B は、記憶装置 6 0 0 を管理する S N 3 0 1 とユーザとのペア毎に課金情報のレコードを作成しサーバ単位課金情報テーブル 2 2 0

に登録することで、サイト毎あるいはSN301毎の課金情報を収集する。

#### 【0037】

本実施形態では、CN101は、ファイル属性や課金情報をリレーショナルデータベースの形式で管理する。

#### 【0038】

図2は、ファイル属性DB110が有する各テーブルの構成例を示す図である。ファイル属性テーブル120には、すべてのファイルに共通するファイル属性が格納される。ファイル属性テーブル120は、ファイルの識別子が登録されるFILE-ID2210、ファイル所有者のIDが格納されるOWNER2220、ファイルの作成日付が格納されるDATE2230、ファイルサイズが格納されるSIZE2240及びファイルの種類を示す情報が格納されるTYPE2250のエントリで構成されている。ファイル属性テーブル120では、1つのファイルに対して1つのレコードが存在する。

#### 【0039】

ACLテーブル130には、ファイルに対するユーザ毎のアクセス権限に関する情報が格納される。具体的には、ACLテーブル130は、ファイルの識別子が登録されるFILE-ID2110、ユーザの識別子が登録されるUSR2120、アクセス権限の種別、例えばファイルの読み出し又は書き込みについてのアクセス権限の有無の情報が登録されるREAD2130及びWRITE2140並びにファイルに対するアクセス可能期限の情報が登録されるEXPIRE2150の各エントリを有する。ACLテーブル130には、ファイルとユーザの組み合わせに対応するレコードが、ファイルとユーザの組み合わせ数分存在する。

#### 【0040】

拡張属性テーブル140は、ファイルの種類に応じた属性を管理する属性テーブルとして使用される。図2では、拡張属性テーブル140を暗号化ファイルの属性を管理するための属性テーブルとして使用する例を示している。この場合、拡張属性テーブル140は、ファイルの識別子が登録されるFILE-ID2410及び暗号化キーの情報が登録される暗号化キー2420のエントリで構成さ

れる。

#### 【0041】

ここで、本計算機システムで管理されるファイルが暗号化ファイルの場合には、図2に示す拡張属性テーブル140が使用される（拡張属性テーブル140にそのファイルが登録される）が、他の種類のファイルでは拡張属性テーブル140は使用（登録）されない。このように、ファイル属性を格納する拡張属性テーブル140をあるファイル属性ごとに個別に設けることで、特定の種類のファイルを特定のテーブルで管理する。このように管理することで、無駄にテーブルのサイズを増大させることなく多種多様なファイル属性を管理することが可能となる。

#### 【0042】

ロケーションテーブル150には、ファイルデータの格納位置を示す情報が保持される。ロケーションテーブル150は、ファイルの識別子が登録されるFILE-ID2310及びファイルデータを格納した場所が登録されるLocation2320のエントリを有する。

#### 【0043】

以上述べた、ファイル属性テーブル120、ACLテーブル130、拡張属性テーブル140及びロケーションテーブル150にはすべてFILE-IDのエントリが含まれている。したがって、各テーブルは、FILE-IDのエントリを用いて相互に関連するリレーショナルデータベース構造となっており、必要に応じて必要な属性を検索することが可能である。

#### 【0044】

図3は、サーバ単位課金情報テーブル220の構成を示した図である。サーバ単位課金情報テーブル220は、ユーザ識別子が登録されるUSR3110、SN301で実行されるFSVR300の識別子が登録されるFSVR-ID3120、FSVR300の実行によりSN301で管理される記憶装置600に格納されているファイルの総記憶容量が登録される使用容量3130、記憶装置600に格納されているファイルの数が登録される総ファイル数3140、記憶装置600からユーザが読み出したデータの量が登録されるRサイズ3150及び

記憶装置 600 にユーザが書き込んだデータの量が登録される W サイズ 3160 の各エントリで構成される。

#### 【0045】

尚、サーバ単位課金情報テーブル 220 は、ユーザのローカル F S の使用情報を SN301（具体的には、SN301 で実行される FSVR300）毎に蓄積するため、各々のユーザと SN301 のペア毎のレコードを有する。

#### 【0046】

CN101B は、MSVR100B を実行し、USR3110 に登録された情報をキーにして課金情報 DB210 内を検索することで、各 SN301 が管理する記憶装置 600 のユーザ毎の使用状況を取得することができる。

#### 【0047】

なお、本実施形態では、リレーショナルデータベースを用いてファイル属性 DB110 及び課金情報 DB210 を管理する例を説明したが、リレーショナルデータベースの代わりにオブジェクト指向データベースや XML データベースを用いることも可能である。

#### 【0048】

次に、本計算機システムでユーザがファイルを読み出すあるいはファイルを更新する（ファイルにデータを書き込む）処理について説明する。

#### 【0049】

まず、ユーザがファイルを更新する場合について説明する。以下、すでにファイル（以下「FILE1」）が本計算機システム内に作成されているとして説明を行う。クライアント 800A を使用するユーザ（識別子として USR1 が与えられている）が FILE1 のデータの更新を行う場合、始めにクライアント 800A が AGENT810 を実行して、CN101A に対してファイルの更新要求（以下「WRITE 要求」）4100 を送信する。

#### 【0050】

図 4 の上段に、クライアント 800A から CN101A に送信される WRITE 要求の具体例を示す。WRITE 要求 4100 には、コマンド種類 4110、ユーザ識別子 4120、ファイル識別子 4130、WRITE 開始オフセット 4



140、データサイズ4150およびデータ4160の各エントリが含まれる。

#### 【0051】

本実施形態のWRITE要求の各エントリには、具体的には以下のような情報が登録される。コマンド種類4110エントリには本要求がデータ更新要求であることを示す「WRITE」、ユーザ識別子4120エントリには要求を発行したユーザを示す情報である「USR1」、及びファイル識別子4130エントリには対象となるファイルを指定する情報である「FILE1」が登録されている。

#### 【0052】

また、WRITE開始オフセット4140、データサイズ4150およびデータ4160エントリには、ファイル内でのデータを更新すべき位置、データサイズ及び実際のデータが各々格納される。

#### 【0053】

図7は、WRITE要求4100を受信したCN101Aの処理手順を示すフローチャートである。CN101Aは、MSVR100Aを実行することで本処理を実行する。

#### 【0054】

WRITE要求4100を受信したCN101Aは、ユーザのアクセス権限を確認するために、受信したファイル識別子であるFILE1及びユーザ識別子であるUSR1をキーとして、記憶装置600Aに格納されたACLテーブル130を検索する検索要求を作成する。CN101Aは、作成された検索要求を実行するためにDBMS102Aの実行を開始する。

#### 【0055】

ここで、図11に、この検索要求の具体例であるACLテーブル検索要求6000Aの内容を示す。

#### 【0056】

ACLテーブル検索要求6000は、コマンド名6001、選択フィールド名6002、検索対象テーブル名6003及び検索条件6004エントリで構成される。図11の例では、コマンド名6001Aに“SELECT”、選択フィー

ルド名 6002A に “WRITE”、検索対象テーブル名 6003A に “ACL テーブル”、及び検索条件 6004A に “FILE-ID=FILE1 AND OWNER=USR1 AND CURDATE () < EXPIRE” が指定されている。

#### 【0057】

上述の登録内容によって、具体的には、ACL テーブル 130 から、FILE-ID が FILE1、OWNER フィールドが USR1、及び EXPIRE フィールドの期限が現在時刻よりも大きい値が登録されているレコードの検索が指定される。

#### 【0058】

CN101A は、DBMS102A を実行することで、ACL テーブル検索要求 6000 で指示された検索処理をファイル属性 DB110 に対して指示し、その結果を記憶装置 600A から受信する。ここで、CN101A が DBMS102A を実行することで実行する検索処理には、データベースの高速検索技術が用いられる。例えば USP6,353,820B1 に記述されているような方法を用いて、高速な検索処理を実行することが可能である。

#### 【0059】

本実施形態では、ACL テーブル検索要求 6000 の検索条件 6004 を満たすレコードがレコード 2111 であるので、検索結果として、レコード 2111 の WRITE フィールド 2140 に登録された値が CN101A に送信される（ステップ 5000）。

#### 【0060】

その後、CN101A は、送信された WRITE フィールド 2140 の値が 1 かどうかにより WRITE 要求が許可されているかどうかを判断する。本実施形態では、値が 1 である場合にはファイルへのアクセス許可を示す。したがって、抽出された WRITE フィールド 2140 の値は 1 であるため、CN101A は、本 WRITE 要求が許可されていると判断する（ステップ 5020）。

#### 【0061】

WRITE フィールド 2140 が 0 の場合は、CN101A は、WRITE 要

求が許可されていないと判断し、クライアント800Aにアクセス不許可の旨を通知して処理を終了する（ステップ5100）。

#### 【0062】

WRITE要求が許可される場合、次にCN101Aは、更新すべきファイルの位置を把握する必要がある。そのため、CN101Aは、ロケーションテーブル検索要求6100作成し、DBMS102Aを実行してロケーションテーブル150を検索することにより、ファイルを更新する位置を示す情報が含まれるロケーション情報を記憶装置600Aから取得する。

#### 【0063】

図11に、ロケーションテーブル検索要求6100の例を示す。FILE1の格納位置を得るためのロケーションテーブル検索要求6100には、検索コマンド6101に“SELECT”、取得フィールド6102に“LOCATION”、検索対象テーブル6103に“ロケーションテーブル”、検索条件6104に“FILE-ID=FILE1”の情報が登録される。

#### 【0064】

このロケーションテーブル検索要求6100を用いて、DBMS102Aを実行するCN101Aは、ロケーションテーブル150の検索を記憶装置600に指示し、FILE-IDフィールドに“FILE1”が指定されているレコード2311及び2312についての情報を記憶装置600Aから取得する。

#### 【0065】

その後、CN101Aは、取得されたレコードから、LOCATIONフィールド2320に格納されている情報である“FSVR1: /FILE1”と“FSVR3: /FILE1”の情報を抽出する（ステップ5030、5040）。

#### 【0066】

LOCATIONフィールド2320に格納された値を抽出したCN101Aは、抽出した値に含まれる全てのSN301に対して、ローカルファイルWRITE要求を送信する。

#### 【0067】

図5の上段に、ローカルファイルWRITE要求4500の具体例を示す。ロ

ーカルファイルWRITE要求4500には、送信先FSVR4510、ローカルファイルI/Oコマンド4520、ローカルファイル名4530、OFFSET4540、SIZE4550及びデータ4560のエントリが含まれる。

#### 【0068】

本実施形態の場合、送信先FSVR4510にはSN301A又はCを示す情報が登録され、ローカルファイルI/Oコマンド4520には、ローカルファイル更新要求であることを示すWRITEが登録される。また、他のエントリには、各々、ローカルファイルの位置等を示す情報が登録される（ステップ5050）。

#### 【0069】

図8は、ローカルファイルWRITE要求4500を受信したSN301A（及び301C）の処理手順を示すフローチャートである。ローカルファイルWRITE要求4500を受信したSN301Aは、初めに記憶装置600Bに格納されたメタデータ330Aを参照し、受信したローカルファイルWRITE要求で指定されたローカルファイル4530が使用する記憶装置600の容量（以下、「使用ディスクサイズ」）を取得する。上述したように、メタデータ330Aには、ローカルFS310Aに格納されたローカルファイルの位置、データ量に関する情報に加え、使用ディスクサイズの情報も含まれている（ステップ5200）。

#### 【0070】

次に、SN301Aは、ローカルファイルシステム310Aに受信したデータを書き込むだけの空き容量が十分にあるかどうかをチェックし、不十分な場合はステップ5230に進みCN101Aにエラーを返す。

#### 【0071】

次に、SN301Aは、受信したローカルファイルWRITE要求のOFFSET4540に登録された情報で示される位置から、受信したデータをSIZE4550で指定されたサイズだけSN301Aが有するキャッシュメモリに書き込むWRITE処理及びそのWRITE処理の結果を記憶装置600Bに反映するSYNC処理を行う（ステップ5210）。

## 【0072】

SYNC処理の完了後、SN301Aは、メタデータ330Aを参照して、書き込み完了後のローカルファイルの使用ディスクサイズを取得する。その後、SN301Aは、ローカルファイルの書き込み前後の記憶装置800の使用ディスクサイズを比較して、ローカルファイルの書き込みによって新たにローカルファイルに与えられた記憶容量、即ち新規ディスク割り当てサイズを算出する（ステップ5220）。

## 【0073】

尚、この新規ディスク割り当てサイズは、サーバ単位課金情報テーブル220の更新時に使用される。サーバ単位課金情報テーブル220に存在する使用容量3120フィールドには、ユーザに使用されている記憶装置600の容量を正確に反映する必要がある。

## 【0074】

このため、各ローカルファイルへのWRITE処理で新規に割り当てられた記憶容量を把握する必要があるが、ローカルファイルに対して上書きを行う場合には、既にローカルファイルが存在しているので、新規記憶容量の割り当て処理が発生せず、使用記憶容量の増分が把握されない。このため、ローカルファイルWRITE処理の前後でWRITE対象のローカルファイルの使用ディスクサイズを比較し、新規ディスク割り当てサイズを計算する。

## 【0075】

最後にSN301Aは、ローカルファイルWRITE処理の結果をCN101Aに送信する。CN101Aに送信される内容には、ローカルファイルWRITE処理に伴うデータの更新が成功したかどうかを示すステータス、書き込み処理が完了したデータのサイズを示す書き込み完了サイズ及び書き込み完了サイズの内新規に割り当てられた新規ディスク割り当てサイズの情報が含まれる（ステップ5230）。

## 【0076】

図7に戻って説明を続ける。

## 【0077】

SN301A (又は301C) から書き込み結果を受信した (ステップ5060) CN101Aは、受信した結果に含まれるステータスの情報をチェックする (ステップ5070)。

#### 【0078】

書き込みに失敗したSN301が存在する場合には、CN101Aは、失敗したSN301が管理する記憶装置600に格納されたローカルファイルの全データの読み出し要求をSN301に送信する。SN301からこれらのデータを取得した後、CN101Aは、取得したデータを書き込みに成功したSN301に送信し、書き込みに成功したSN301が管理する記憶装置600に格納されたローカルファイルの書き換えを指示する。これにより、CN101Aは、計算機システム内で分散された全てのローカルファイルを書き込み前の状態に戻す。その後、CN101Aは、クライアント800Aにエラーを示す結果を送信する (ステップ5075)

全てのローカルファイルに対するデータの書き込みに成功した場合、CN101Aはファイル属性テーブル120の更新処理を行う。具体的には、CN101Aは、ファイル属性テーブル120内のファイル識別子4130で示されるファイルに対応するレコードのSIZEフィールド2240の値をWRITE実行後のサイズに更新する。

#### 【0079】

まず、CN101AはSIZEフィールド取得要求を作成する。この要求には、検索コマンド“SELECT”、検索先テーブルにファイル属性テーブル、取得フィールドにSIZE、検索条件としてFILE-IDフィールドがファイル識別子4130と等しい条件を指定する情報が含まれる。CN101AがDBMS102Aを実行してこのSIZEフィールド取得要求を処理することで、WRITE実行前のSIZEフィールド2240の値が取得できる。

#### 【0080】

次に、CN101Aは、WRITE要求4100のOFFSET4140とSIZE4150に登録された値を足した値と、SIZEフィールド取得要求で取得されたSIZEフィールド2240に登録された値を比較する。SIZEフィ

ールド取得要求で取得された値が足された値と等しい又は大きい場合は、ファイルサイズの更新処理の必要が無いため、CN101Aは、本ファイル属性テーブル更新処理を終了し、次の処理を実行する。

#### 【0081】

SIZEフィールド取得要求で取得された値が足された値より小さい場合には、CN101Aは、OFFSET4140とSIZE4150を足した値を新しいファイルサイズとしてファイル属性テーブルを更新する。

#### 【0082】

具体的には、CN101Aはファイルサイズ更新要求を作成する。ファイルサイズ更新要求には、更新コマンド“UPDATE”、検索先テーブルにファイル属性テーブル、更新命令として“SET SIZE=新ファイルサイズ”、更新条件としてFILE-IDフィールドがファイル識別子4130と等しい条件を指定する情報が含まれる。CN101Aが、DBMS102Aを実行してこのファイルサイズ更新要求を処理することにより、ファイル属性テーブルが更新され、結果が記憶装置600に反映される（ステップ5075）。

#### 【0083】

ファイル属性テーブル更新処理が終了すると、CN101Aは、CN101Bに課金情報DB更新要求4700を送信する（ステップ5080）。

#### 【0084】

図6に、課金情報DB更新要求4700の内容の具体例を示す。課金情報DB更新要求4700は、送信先MSVR4710、ユーザ識別子USR4720、ファイルサーバ識別子4730及びファイルサーバ毎書き込み情報リスト4740のエントリを有する。

#### 【0085】

ファイルサーバ毎書き込み情報リストエントリ4740は更に、ファイルサーバ識別子4750、READ完了サイズ4760、WRITE完了サイズ4770及びディスクサイズ4780のエントリを含む。

#### 【0086】

CN101Aより課金情報DB更新要求を受信したCN101Bは、課金情報

テーブル 220 の更新処理を行う。課金情報テーブル 220 の更新処理については後述する（ステップ 5080）。

#### 【0087】

CN101B から課金情報 DB 更新処理終了の通知を受信した CN101A は、クライアント 800A に対して書き込み完了を報告し（ステップ 5090）、WRITE 処理を終了する（ステップ 5100）。

#### 【0088】

図 9 は、課金情報 DB 更新要求を受信した CN101B が行う課金情報テーブル 220 の更新処理の手順を示す図である。

#### 【0089】

CN101B は最初に、ユーザ識別子及び SN301 を示す識別子のペアをキーにして、課金情報テーブル 210 内でそれらペアと一致するレコードの検索並びに検索されたレコードに含まれる使用容量 3120 エントリ及び W サイズ 3150 エントリにそれぞれ課金情報 DB 更新要求に含まれるディスクサイズ 4780 エントリと WRITE サイズ 4770 エントリに登録された情報の足しこみを指示するデータベース更新要求 6200 を作成し、DBMS102B の実行を開始する。

#### 【0090】

データベース更新コマンド要求は、コマンド名 6201、更新テーブル名 6202、更新情報 6203 及び検索条件 6404 のエントリを有する。図 12 は、データベース更新要求 6200 の具体例を示した図である。データベース更新要求 6200A（及び B）の各エントリには、コマンド名 6201A（及び 6201B）に“UPDATE”を、更新テーブル名 6202A（及び 6202B）に“課金情報テーブル”を、検索条件 6204A（及び 6204B）に“USR=USR1 AND FSVR-ID=FSVR300A”を示す情報が指定される。また、更新情報 6203A には“使用容量=使用容量+DISK サイズ”を示す情報が、更新情報 6203B には“W サイズ=W サイズ+WCSIZE”を示す情報が指定されている。

#### 【0091】



本実施形態の場合、USR1及びFSVR1のキーで選択される課金情報DB 210内のレコード3111並びにUSR1及びFSVR3のキーで選択されるレコード3112の2つのレコードが存在する。このため、始めにCN101Bは、USR1及びFSVR1のキーで選択されるレコード3111に対するデータベース更新コマンド6200A（および6200B）を作成し、DBMS102Bを実行する（ステップ5400）。

#### 【0092】

DBMS102Bを実行するCN101Bは、作成されたデータベース更新コマンド6200A（および6200B）に基づいて、各々のレコードの使用容量3120エントリ及びWサイズ3150エントリに、新規割り当てディスクサイズと書き込み完了サイズを足しこんだ結果を記憶装置600Aに書き込む処理を行う。

#### 【0093】

データベース更新要求処理時に、指定したユーザ識別子とFSVR300を指定する識別子の組に一致するレコードが存在しない場合、課金情報DB更新処理はエラーとなる。このため、CN101Bは、データベース更新要求6200の処理が成功したかどうかについての情報（実行結果）を記憶装置600Aから収集する。

#### 【0094】

CN101Bは、収集されたデータベース更新要求実行結果を参照し、本処理が成功したかどうかを判断する（ステップ5410）。

#### 【0095】

USR1及びFSVR1のキーに基づいた課金情報DB更新処理が失敗した場合、それはUSR1及びFSVR1をキーとしたレコードが、課金情報テーブル220に存在しないことを意味する。このため、このエラーが発生した場合には、CN101Bは、新規レコード追加要求を記憶装置600Aに送信する必要がある。尚、このエラーは、新規ファイルに対する書き込みの場合に発生するが、ここでは既存のファイルに対する書き込み処理の例であるため発生しない。本処理の詳細は、後述の新規ファイル作成処理で説明する（ステップ5420）。

**【0096】**

CN101Bは、USR1及びFSVR1に対するデータベース更新処理要求が完了すると、課金情報DB更新要求4700のファイルサーバ毎書き込み情報リスト4740に指定されたSN301全てに対する処理が終了したかどうか判断し（ステップ5430）、終了していない場合にはステップ5400～5420を繰り返す。本実施形態の場合、CN101Bは、次のUSR1及びFSVR3をキーとしたレコードに対し、ステップ5400～ステップ5420の処理を繰り返す。

**【0097】**

課金情報DB更新要求に含まれるファイルサーバ毎書き込み情報リスト4740に指定された全SN301に対する課金情報更新処理が終了すると、CN101Bは、結果をCN101Aに送信し、処理を終了する（ステップ5440）。

**【0098】**

このように、データベースで課金情報を管理することで、SN301毎の課金情報を容易に管理することが可能となる。

**【0099】**

尚、本実施形態では、ファイル属性DB110と課金情報DB210を異なるCNR101で管理する例を説明したが、両DBをひとつのCN101で管理する構成としても良い。

**【0100】**

次に、ユーザがクライアント800Aを介してファイルを読み出すREAD処理の例を説明する。尚、ここでは、すでにあるファイル（以下「FILE2」）がシステム内に作成されているとして説明を行う。

**【0101】**

クライアント800Aを使用するユーザ（以下「USR2」）がFILE2の読み出しを行う場合、始めにクライアント800Aは、AGENT810を実行して、CN101Aに対してREAD要求4200を送信する。

**【0102】**

図4の中段に、READ要求の具体例を示す。READ要求4200には、コ

マンド種類 4210、ユーザ識別子 4220、ファイル識別子 4230、READ 開始オフセット 4240 及びデータサイズ 4250 の各エントリが含まれる。本例では、コマンド種類 4210 エントリには、ファイルの読み出し要求であることを示す「READ」が、ユーザ識別子 4220 には「USR2」が、ファイル識別子 4230 には、「FILE2」を示す情報が登録されている。また、READ 開始オフセット 4240、データサイズ 4450 には、ファイル内でのデータを読み出すべき位置、データサイズが各々格納される。

#### 【0103】

なお以下では特に断りのない限り、CN101A は MSVR100A を実行することで処理を行う。

#### 【0104】

READ 要求 4200 を受信した CN101A は、ユーザのアクセス権限を確認するために、受信したファイル識別子である FILE2 及びユーザ識別子である USR2 をキーとして、ACL テーブル 130 を検索する ACL テーブル検索要求 6000B を作成する。次に、DBMS102A を実行して上記 ACL テーブル検索要求 6000B を記憶装置 600A に送信し、アクセス権限に関する情報を記憶装置 600A から取得する。

#### 【0105】

ACL テーブル検索要求 6000B の具体例を図 11 に示す。ACL テーブル検索要求 6000B の各エントリには、コマンド名 6001B に“SELECT”を、選択フィールド名 6002B に“READ”を、検索対象テーブル名 6003B に ACL テーブルを、検索条件 6004B に“FILE-ID=FILE2 AND OWNER=USR2 AND CURDATE () < EXPIRE”を指定する情報が登録されている。

#### 【0106】

この ACL テーブル検索要求 6000B により、CN101A は DBMS102A を実行して、ACL テーブルから FILE-ID に FILE2 を持ち、OWNER フィールドに USR2 を持ち、EXPIRE フィールドの期限が現在時刻よりも大きいレコードを選択し、その READ フィールドを記憶装置 600A か

ら取得する。

【0107】

本実施形態では、検索条件6004Bエントリに登録された条件を満たすレコードはレコード2113であるため、ここでは、CN101Aは、記憶装置600Aから、レコード2113のREADフィールド2130に登録された値を取得できる。

【0108】

CN101Aは、DBMS102Aの実行により得られたREADフィールドの値が1かどうかによりREAD要求が許可されているかどうかを判断する。ここでは、READフィールド2130の値は1であるため、CN101Aは本READ要求が許可されていると判断する。

【0109】

READフィールド2130が0の場合は、CN101Aは本WRITE要求が許可されていないと判断し、クライアント800Aにアクセス不許可の旨を通知して処理を終了する。

【0110】

READ要求が許可される場合、CN101Aは読み出すべきファイルの位置を把握する必要がある。CN101Aは、読み出すべきファイルの位置についての情報であるロケーション情報を、WRITE処理と同様にロケーションテーブル検索要求6100をDBMS102Aを実行して処理することにより記憶装置600から取得する。この時、ロケーションテーブル検索要求6100の検索条件6140には“FILE-ID=FILE2”が指定される。

【0111】

本実施形態においては、この処理により“FSVR1:／FILE2”と“FSVR3:／FILE2”の2つのロケーション情報が取得される。

【0112】

次にCN101Aは、ファイル識別子であるFILE2を用いて、ファイル属性テーブル120を検索し、条件を満たすレコードを記憶装置600Aから読み出す。そして、CN101Aは、選択されたレコードのファイルサイズ2240

及びファイルタイプ 2250 エントリに登録されている情報を取得する。本実施形態では、CN101A は、選択されたレコード 2212 から、ファイルサイズ 20MB 及びファイルタイプ ENCRYPT という情報を取得する。

#### 【0113】

ここで、ENCRYPT という情報は、CN101A がクライアント 800 にファイルを転送する時に、そのファイルを暗号化して送信しなければならないファイルタイプであることを意味する。したがって、クライアント 800A へファイルデータを転送するときに、CN101A は、拡張属性テーブル 140 に格納されている暗号化キーを用いてファイルデータを暗号化する必要がある。

#### 【0114】

本実施形態においては、ファイルデータが格納されている SN301 が管理する記憶装置 600 のいずれからでもデータを取得できるため、CN101A は、SN101A 又は SN101C の何れか、もしくは両方にローカルファイル READ 要求 4600 を発行することができる。

#### 【0115】

以下、CN101A が、CN101A と同じネットワーク 700 に接続されている SN301A にローカルファイル READ 要求を送信する例を説明する。ただし、CN101A が、SN301A と SN301C の両方に対し、ローカルファイルから半分ずつのデータを読み出すローカルファイル READ 要求を発行しても良い。

#### 【0116】

図5の中段に、ローカルファイル READ 要求 4600 の具体例を示す。ローカルファイル READ 要求 4600 には、送信先 FSVR 4610、ローカルファイル I/O コマンド 4620、ローカルファイル名 4630、OFFSET 4640 及び SIZE 4650 の各エントリが含まれる。ローカルファイル I/O コマンド 4620 エントリには、ファイルの読み出しを示す「READ」の情報が登録される。また、ローカルファイル名 4630 エントリには、FILE 2 を示す「FILE 2」が登録される。

#### 【0117】

なお以下では、特に断りのない限り、SN301Aの処理は、FSVR300Aの実行により行われる。

**【0118】**

ローカルファイルREAD要求4600を受信したSN301Aは、指定されたローカルファイルのOFFSET4640エントリに登録された位置からSIZE4550エントリで指定されたサイズのデータを読み出すローカルファイルREAD処理を行う。具体的には、SN301Aは必要なファイルデータを取得するため、適宜記憶装置600Bに読み出し要求を発行する。

**【0119】**

SN301Aは、ローカルファイルREAD処理が終了したら、読み出したデータとサイズをCN101Aに送信する。

**【0120】**

CN101Aは、SN301AからローカルファイルREAD要求の結果を受信すると、受信した結果に含まれるステータスをチェックし、ローカルファイルの読み出しが成功したかどうかを確認する。

**【0121】**

SN301AのローカルファイルREAD処理が失敗した場合には、SN301Cに対して同様の要求を発行する。このように、複数のローカルファイルが存在する場合、あるSN301の読み出しが失敗しても、他のSN301を用いて読み出し処理を行うことが可能であり、信頼性が向上する。ロケーションテーブル150に登録されている全てのSN301の読み出しが失敗した場合には、CN101Aはクライアント800に対してエラーを返す。

**【0122】**

SN301Aからのローカルファイルの読み出しが成功すると、CN101Aは課金情報DB更新要求4700を作成し、CN101Bに送信する。以下、特に断りのない限り、CN101BはMSVR100Bを実行して処理を行う。

**【0123】**

課金情報DB更新要求4700を受信したCN101Bは最初に、ユーザ識別子及びFSVR300を示す識別子のペアをキーにして、課金情報テーブル21

0内でそれらペアと一致するレコードの検索及び検索されたレコードに含まれる R サイズ 3140 エントリに、課金情報 DB 更新要求 4700 に含まれる READ 完了サイズ 4760 エントリに登録された情報の足しこみを記憶装置 600A に指示する課金情報テーブル更新要求 6200C を作成する。

#### 【0124】

課金情報テーブル更新要求 6200C は、コマンド名 6201、更新テーブル名 6202、更新情報 6203 及び検索条件 6404 の各エントリを有する。図 12 ではコマンド名 6201C に “UPDATE” を、更新テーブル名 6202C に “課金情報テーブル” を、更新情報 6203C には “R サイズ = R サイズ + R C S I Z E” を、検索条件 6204C に “USR = USR2 AND F S V R - I D = S N 3 0 1 A” を指定する情報が登録されている例を示す。

#### 【0125】

課金情報テーブル 220 で USR2 及び SN301A の組に該当するレコードは、レコード 3113 である。このため、CN101B は、DBMS102B を実行して、レコード 3113 の R サイズ 3140 エントリに READ 完了サイズを足しこんだ結果を書き込む処理を、課金情報テーブル更新要求 6200C で記憶装置 600A に指示する。

#### 【0126】

CN101B は、次に課金情報 DB 更新要求 4700 のファイルサーバ毎書き込み情報リスト 4740 エントリに指定された全てのファイルサーバに対する処理が終了したかどうか判断し、終了していない場合には前述のデータベース更新要求処理を繰り返す。本実施形態の場合、他に指定されたファイルサーバが存在しないため、CN101B はこの時点で課金情報更新処理を終了し、結果を CN101A に送信する。

#### 【0127】

CN101B から課金情報 DB 更新処理終了の通知を受信した CN101A は、ファイル識別子である FILE2 をキーにして記憶装置 600A 内の拡張属性テーブル 140 を検索し、FILE2 が拡張属性テーブル 140 に登録されているかを確認する。本例の場合、CN101A は、KEY2420 エントリより、

FILE 2 をクライアント 800 に送信するために必要な暗号化キー KEY 1 を取得する。尚、拡張属性テーブル 140 にファイルが登録されていない場合、この属性に対応する処理（本例では暗号化）は行われない。

#### 【0128】

その後、CN101A は、KEY 1 を用いて読み出されたファイルデータを暗号化し、クライアント 800 に対して読み出し結果として暗号化されたファイルデータを送信し、READ 処理を終了する。

#### 【0129】

図 10 は、計算機システム内に新しいファイル（以下「新規ファイル」）を作成する場合の処理手順を示す図である。以下の処理は、CN101A が MSVR 100A を実行して行う処理である。

#### 【0130】

新規ファイルを作成する場合、ユーザの指示を受けたクライアント 800 は AGENT プログラム 810A を実行して、CN101A に対して CREATE 要求 4300 を送信する。図 4 の下段に CREATE 要求 4300 の例を示す。CREATE 要求 4300 には、ファイル作成コマンド 4310、ユーザ ID 4320、ファイルの冗長度 4330、ファイルタイプ 4340、暗号化キー 4350、及び有効期限 EXP\_DATE 4360 のエントリが含まれる。

#### 【0131】

CREATE 要求 4300 を受信した CN101A は、ファイルに割り当てるファイル ID を取得する。この処理には、ファイル ID ビットマップ 160 が用いられる。ファイル ID ビットマップ 160 には、計算機システムで使用可能なファイル ID が登録され、さらに計算機システムで使用中のファイル ID にはビット 1 が、未使用のファイル ID にはビット 0 が割り当てられている。

#### 【0132】

CN101A は、記憶装置 600A に格納されたファイル ID ビットマップ 160 を検索して 0 のビットを探すことで、未使用のファイル ID を取得することができる。0 のビットが見つかったら、使用中であることを示すために CN101A はそのビットを 1 にするよう、記憶装置 600 に指示する（ステップ 5600



)。

#### 【0133】

次にCN101Aは、ファイルデータを格納するSN301を選択する。CREATE要求4300には、ファイルの冗長度が指定されており、CN101Aは指定された冗長度に相当する数のSN301を選択する。この選択は、たとえばCN101Aがネットワーク700にブロードキャストメッセージを流し、最初にリプライを返したSN301から順に選択する方法を使用する(ステップ5610)。

#### 【0134】

次にCN101Aは、ステップ5610で選択されたSN301にローカルファイル作成要求4690を送信し、その実行結果を受け取る(ステップ5620)。図5の下段に、ローカルファイル作成要求4690の例を示す。ローカルファイル作成要求4690には、ファイルを作成するSN301名FSVR4691、ローカルファイル作成コマンドCREATE4692及びローカルファイル名4693のエントリを有する。ここで、ローカルファイル名4693には、ステップ5600で選択したファイルIDを示す情報が登録される。

#### 【0135】

ローカルファイル作成要求4690を受信したSN301はFSVRを実行し、コマンドに指定されたローカルファイル名4693エントリに登録された値を用いてローカルファイルを作成し、その結果をCN101Aに返信する。

#### 【0136】

結果を受信したCN101Aは、受信したローカルファイル作成結果が成功したかどうかを判断し(ステップ5625)、失敗した場合にはステップ5610に戻る。ローカルファイルの作成が成功した場合、CN101Aは、ローカルファイル作成に成功したSN301の名前FSVRをロケーションテーブル150に登録する。この登録処理は、CN101AがDBMS102Aを実行することで実現する。このとき、記憶装置600へは、ロケーション登録要求6110が指定される。

#### 【0137】

図11に、ロケーション登録要求6110の具体例を示す。ロケーション登録要求6110には、登録コマンドINSERT6111、テーブル名6112及び登録するレコードの値6113の各エントリが含まれる。本例では、登録するレコードの値6113に登録される値として、FILE-IDフィールドにステップ5600で割り当てたファイルIDが、LOCATIONフィールドにローカルファイルの作成に成功したSN301名“FSVR”と作成したローカルファイル名がセットされる（ステップ5630）。

#### 【0138】

ロケーションテーブル150への情報の登録が完了すると、CN101Aは、ファイルの冗長度4330で指定された数のローカルファイルの作成が完了したかどうかを調べる（ステップ5640）。冗長度4330で指定された数のローカルファイル作成が完了していない場合には、ステップ5610から再び処理を行なう。

#### 【0139】

一方、指定された数のローカルファイル作成が完了した場合には、CN101Aは、ファイルタイプ4330で指定されたファイルタイプが拡張属性テーブル1140に登録すべきものかどうか、本例ではENCRYPTかどうかを調べる（ステップ5650）。

#### 【0140】

ファイルタイプがENCRYPTである場合、CN101Aは、暗号化キー4350で指定されたキー及びファイルIDを拡張属性テーブル140に登録する処理を行なう。この登録処理は、CN101AがDBMS102Aを実行することで実現され、その際、記憶装置600への指示として、拡張属性登録要求6020が作成される。

#### 【0141】

図11に、拡張属性登録要求6020の例を示す。拡張属性登録要求6020には、DB登録コマンド6021、拡張属性テーブル名6022、登録するレコードの値6023の各エントリが含まれる。登録するレコードの値6023に登録される値としては、FILE-IDフィールドにステップ5600で割り当て

たファイルIDが、KEYフィールドに暗号化キー4350で指定されたキーがセットされる（ステップ5660）。

#### 【0142】

ステップ5650でファイルタイプがENCRYPTでないと判断された場合又はステップ5660で拡張属性テーブルへの値の登録後、CN101Aは、ファイル属性テーブル120の登録処理を行なう。

#### 【0143】

このファイル属性テーブルへの登録処理は、CN101AがDBMS102Aを実行することで実現するが、CN101AはDBMS102Aを実行する前に、DBMS102A実行時に必要なファイル属性テーブル登録要求6030を作成する。

#### 【0144】

図11に、ファイル属性テーブル登録要求6030の例を示す。ファイル属性テーブル登録要求6030には、DB登録コマンド6031、ファイル属性テーブル名6032、登録するレコードの値6033の各エントリが含まれる。登録するレコードの値6033エントリのFILE-IDフィールドにはステップ5600で割り当てられたファイルIDが、OWNERフィールドにはユーザID4320で指定された値が、DATEフィールドには現在の日時を示すCURDATE（）が、SIZEフィールドには0が、及びTYPEフィールドにはファイルタイプ4340で示される値が設定される。

#### 【0145】

CN101Aは、DBMS102A実行時にファイル属性テーブル登録要求6030を参照し、その内容に従ってファイル属性テーブル120への登録処理を行い、結果を記憶装置600に反映する（ステップ5670）。

#### 【0146】

CN101Aは、ファイル属性テーブルへの登録処理が終了すると、新規ファイルのACLテーブルへの登録処理を行なう。この登録処理は、CN101AがDBMS102Aを実行し、以下のACLテーブル登録要求6010を処理することで実現される。

**【0147】**

図11に、ACLテーブル登録要求6010の例を示す。ACLテーブル登録要求6010には、DB登録コマンド6011、ACLテーブル名6012、登録するレコードの値6013のエントリが含まれる。登録するレコードの値6013に登録される値としては、FILE-IDフィールドにステップ5600で割り当てられたファイルIDが、OWNERフィールドにユーザID4320で指定された値が、EXPIREフィールドにEXP\_DATE4360の値がセットされる（ステップ5680）。

**【0148】**

ACLテーブルへの登録処理が完了すると、CN101Aは、クライアント800に対してファイル作成処理が成功したかどうかの情報を返信する。この時、ファイル作成処理に成功した場合には、ファイルIDも同時に返す（ステップ5690）。

**【0149】**

次に、図1および図13を用いてユーザ毎の請求金額を計算する実施形態を説明する。上述したように、本計算機システムは、記憶装置600の使用情報をSN301毎に管理している。このため、SN301毎に課金方針や課金単価が異なる場合でもそれらを反映した課金を行うことが可能となる。

**【0150】**

本実施形態では、計算機システムに新たに課金サーバ900及び課金データDBが格納される記憶装置600Dが追加される。課金サーバ900は、上述したクライアント800等と同様の計算機である。記憶装置600Dには、以下に説明される各種テーブルが格納される。尚、課金サーバ900と記憶装置600Dは同一筐体でも別筐体でも良い。本実施形態では、課金データDB930を用いてユーザごとの請求金額が計算される。

**【0151】**

具体的には、課金サーバ900は、請求金額を計算するために課金計算プログラム910を実行する。また、課金サーバ900は、課金を行うユーザのユーザIDをユーザテーブル940で、SN301毎の課金ポリシーを課金ポリシーテ

ーブル 9 5 0 で管理する。

#### 【0 1 5 2】

さらに課金サーバ 9 0 0 は、記憶装置 6 0 0 D のユーザテーブル 9 4 0 及び課金ポリシーテーブル 9 5 0 を検索し、その検索結果を課金請求テーブル 9 6 0 に格納するためのプログラムである DBMS 9 2 0 を実行する。

#### 【0 1 5 3】

図 1 4 は、ユーザテーブル 9 4 0 の構成例を示す図である。ユーザテーブルは本計算機システムを使用するユーザの情報を保持するテーブルであり、ユーザ ID 7 0 0 0、氏名 7 0 1 0、請求先 7 0 2 0 及び備考 7 0 3 0 等のエントリを含む。。

#### 【0 1 5 4】

図 1 4 上段に、課金請求テーブル 9 6 0 の例を示す。課金請求テーブル 9 6 0 は、一定使用期間毎の請求金額についての情報を保持するテーブルであり、本テーブル 9 6 0 の情報に基づいて、計算機システムの所有者等は使用料を請求する。本テーブル 9 6 0 には、ユーザ ID 7 1 0 0、使用月 7 1 1 0 及び請求金額 7 1 2 0 などのエントリが含まれる。なお本テーブル 9 6 0 には、請求済みかどうか、徴収済みかどうかなどの情報を登録するエントリが含まれていても良い。

#### 【0 1 5 5】

図 1 4 中段に、各ファイルサーバ毎の課金ポリシーを保持する課金ポリシーテーブル 9 5 0 の例を示す。本テーブル 9 5 0 に格納された情報及び課金情報 DB 2 1 0 に格納された情報を元に、課金サーバ 9 0 0 はユーザ毎の請求金額を計算する。本テーブル 9 5 0 には、ファイルサーバ名 7 2 0 0、単位容量あたりの使用金額を示す容量単価 7 2 1 0、ファイル 1 つあたりの使用金額を示すファイル単価 7 2 2 0、単位サイズの読み出しあたりの使用金額を示す READ 単価 7 2 3 0 及び単位サイズの書き込みあたりの使用金額を示す WRITE 単価 7 2 4 0 の各エントリが含まれる。

#### 【0 1 5 6】

図 1 3 は、課金サーバ 9 0 0 が課金計算プログラム 9 1 0 を実行して行う処理手順を示す図である。

**【0157】**

課金サーバ900は、はじめに記憶装置600Dのユーザテーブル940からユーザIDのリストを取得する。この処理は、課金計算サーバ900がDBMS920を実行し、以下のユーザIDリスト取得要求6420を処理することで実現される。

**【0158】**

図12に、ユーザIDリスト取得要求6420の例を示す。ユーザIDリスト取得要求6420では、検索コマンドエントリ6421に“SELECT”、取得フィールド名エントリ6422に“USR”、検索先テーブル名エントリ6423に“ユーザテーブル”を示す情報が登録される（ステップ5600）。

**【0159】**

次に、課金サーバ900は、取得したリストの先頭のユーザIDを取得する（ステップ5610）。

**【0160】**

その後、課金サーバ900は、取得されたユーザIDに基づいて、課金情報取得処理を行なう。この処理は、課金サーバ900がCN101Bに課金情報検索要求6410を発行し、CN101BがDBMS102Bを実行することで実現される。

**【0161】**

図12に、課金情報検索要求6410の例を示す。課金情報検索要求6410には、検索コマンド“SELECT”6411、全フィールド取得指定“\*”6412、検索先テーブル6413、検索条件6414が含まれる。検索条件6414としては、USRフィールド3110の値がステップ5600で取得したユーザIDと等しいレコードを取得する条件が指定される（ステップ5620）。

**【0162】**

課金情報を取得した課金サーバ900は、次にSN301毎の請求金額を計算する。取得された課金情報には、同一ユーザが使用した全SN301に関するサーバ単位課金情報テーブル220のレコードが含まれている。課金サーバ900は、各レコードに含まれるFSVR-IDフィールド3110の値を用いて課金

ポリシテーブル 950 を検索することで、その SN301 の課金ポリシを取得し、取得した課金ポリシとサーバ単位課金情報テーブル 220 のレコードの情報とを用いて SN301 毎の請求金額を計算する。

#### 【0163】

この時、課金ポリシテーブル 950 の検索は、課金サーバ 900 が DBMS 920 を実行し、課金ポリシテーブル検索要求 6430 を処理することで実現される。図 12 に、課金ポリシテーブル検索要求 6430 の構成例を示す。課金ポリシテーブル検索要求 6430 は、検索コマンド “SELECT” 6431、全フィールド取得指定 “\*” 6432、検索先テーブル “課金ポリシテーブル” 6433、検索条件 6434 を含む。検索条件 6434 には、FSVR フィールド 7200 が現在の請求金額計算対象 SN301 の名前 FSVR と等しいという条件が指定される。

#### 【0164】

尚、選択された SN301 に対応する請求金額は、ステップ 5600 で取得された課金情報に含まれる使用容量フィールド 3120、ファイル数フィールド 3130、R サイズフィールド 3140、W サイズフィールド 3150 に登録された値に、それぞれ課金ポリシの容量単価 7210、ファイル単価 7220、READ 単価 7230、WRITE 単価 7240 の値を掛け合わせて計算できる（ステップ 5630）。

#### 【0165】

SN301 毎の請求金額が計算されると、課金サーバ 900 は、全 SN301 の請求金額を合計し、その結果を課金請求テーブル 960 に格納する。この処理は、課金サーバ 900 が DBMS 920 を実行し、課金請求テーブル登録要求 6400 を処理すること出実現される。

#### 【0166】

図 12 に、課金請求テーブル登録要求 6400 の例を示す。課金請求テーブル登録要求 6400 は、登録コマンド “INSERT” 6401、登録先テーブル名 “課金請求テーブル” 6402 及び登録内容 6403 を含む。登録内容 6403 には、ユーザ ID、使用月及び請求金額を示す情報が含まれる（ステップ 56

40)。

【0167】

SN301の請求金額計算を完了した課金サーバ900は、全ユーザの請求金額計算の処理が完了したかどうかを判定する。まだ処理の完了していないユーザが存在する場合には、課金サーバ900は、ステップ5610からステップ5640の処理を繰り返す(ステップ5650)。

【0168】

以上説明したとおり、本実施形態により、SN301毎の課金ポリシーを反映したきめ細かな課金制御が可能となる。

【0169】

【発明の効果】

本発明によれば、ファイル毎に柔軟にファイル属性を付加したり、多数ユーザのアクセス権限情報を効率よく管理する計算機システムを実現することができる。また、複数のサイトに分散したサーバが管理する記憶装置を利用してファイルを格納する場合に、サイトやサーバ毎に課金情報を管理する計算機システムを実現できる。

【図面の簡単な説明】

【図1】

本発明を適用した計算機システムの全体構成の例を示す図である。

【図2】

ファイル属性DBの各種テーブルの例を示す図である。

【図3】

課金情報DBの例を示す図である。

【図4】

ファイルIO要求コマンドの例を示す図である。

【図5】

ローカルファイルIO要求の例を示す図である。

【図6】

課金情報DB更新要求の例を示す図である。



**【図 7】**

MSVR100Aの処理を示すフローチャートである。

**【図 8】**

FSVR300の処理を示すフローチャートである。

**【図 9】**

MSVR100Bの処理を示すフローチャートである。

**【図 10】**

新規ファイルを作成する場合の処理を示すフローチャートである。

**【図 11】**

ACLテーブルの検索要求と登録要求、拡張属性テーブル登録要求及びロケーションテーブルの検索要求と登録要求の例を示す図である。

**【図 12】**

課金情報テーブルの更新要求と登録要求、請求金額の計算処理に使用する各種テーブルの登録要求および検索要求の例を示す図である。

**【図 13】**

課金サーバ900の処理を示すフローチャートである。

**【図 14】**

ユーザテーブル、課金請求テーブル及び課金ポリシーテーブルの例を示す図である。

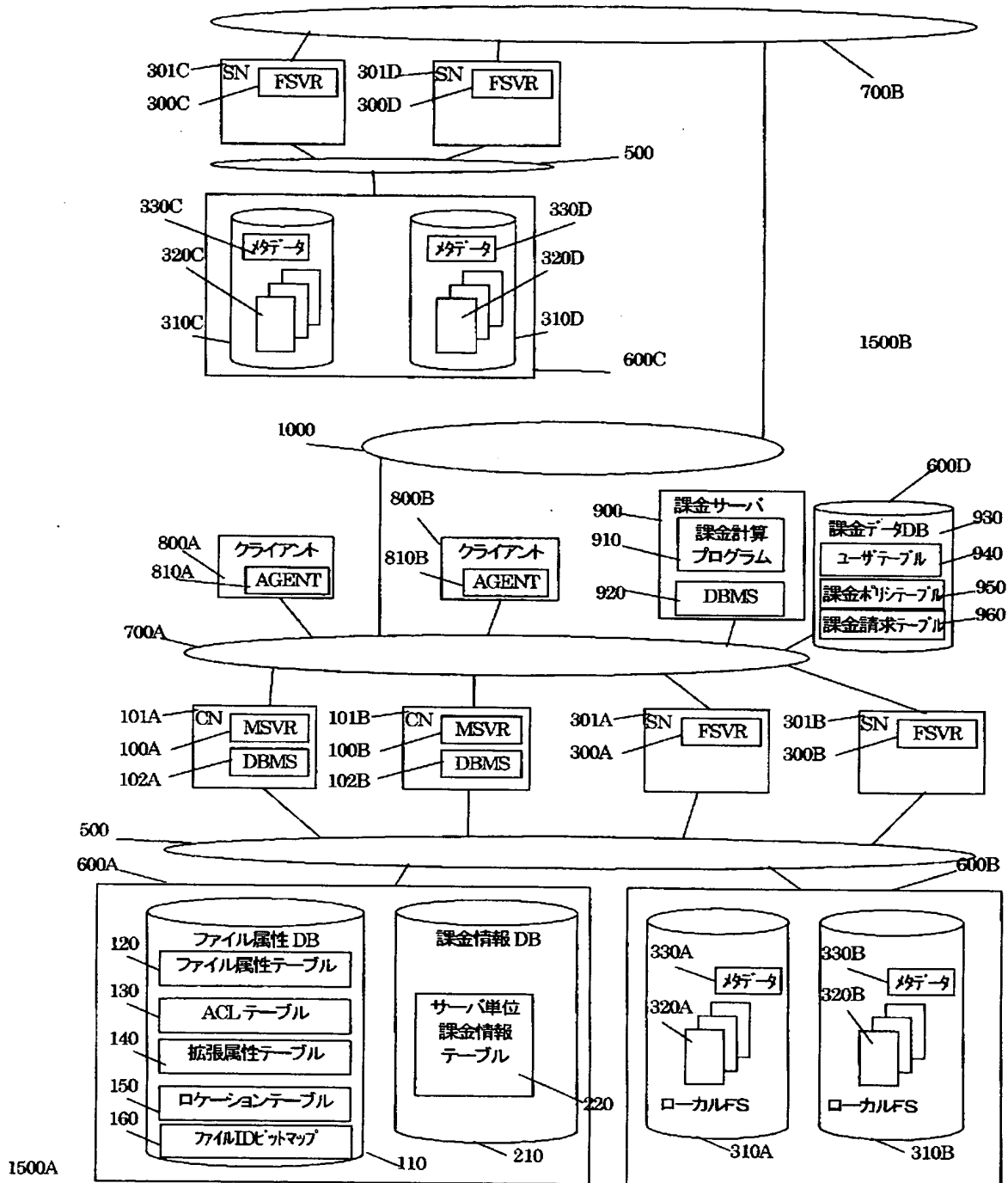
**【符号の説明】**

100…MSVR、300…FSVR、500…ストレージネットワーク、600…記憶装置、700…ネットワーク、1000…インターネット。

【書類名】 図面

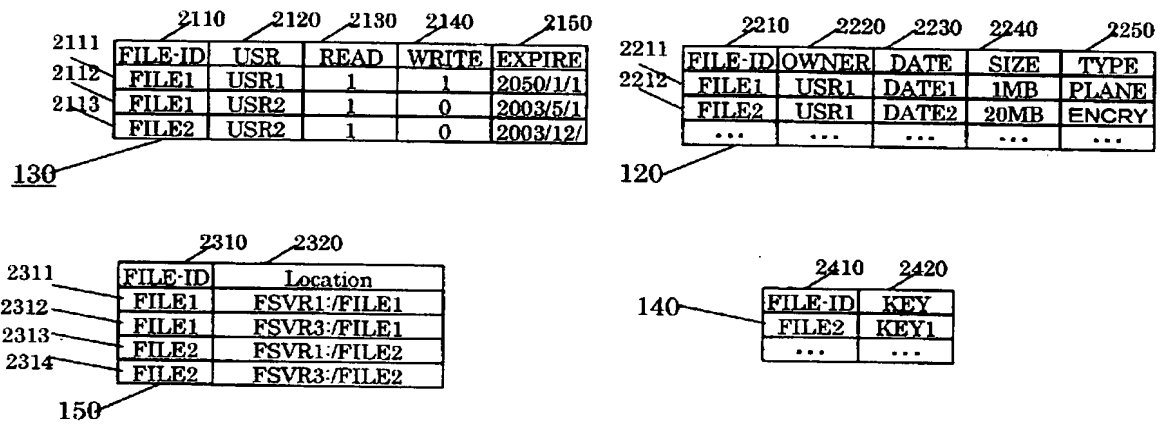
【図 1】

図 1



【図 2】

図 2



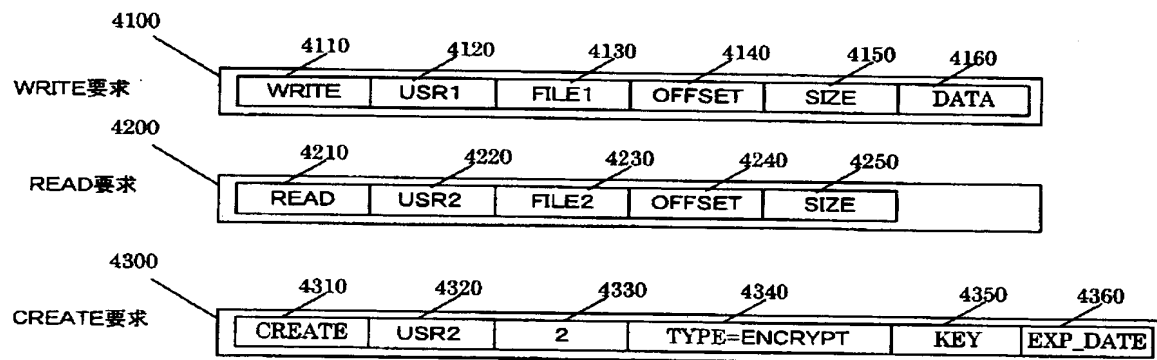
【図 3】

図 3

USR	FSVR-ID	使用容量	ファイル数	R サイズ	W サイズ
USR1	FSVR1	30M	50	500M	30M
USR1	FSVR3	20G	100	100G	20G
USR2	FSVR1	10G	1000	100G	10G

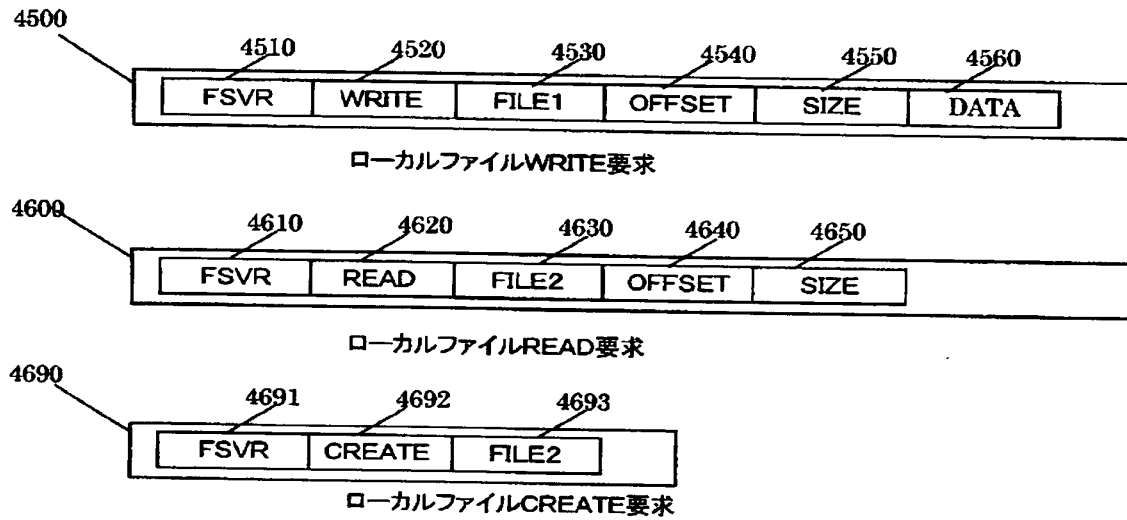
【図 4】

図 4



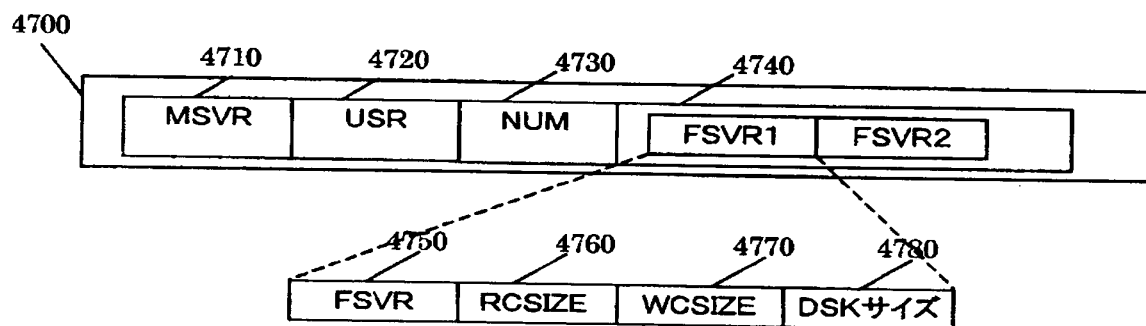
【図 5】

図 5



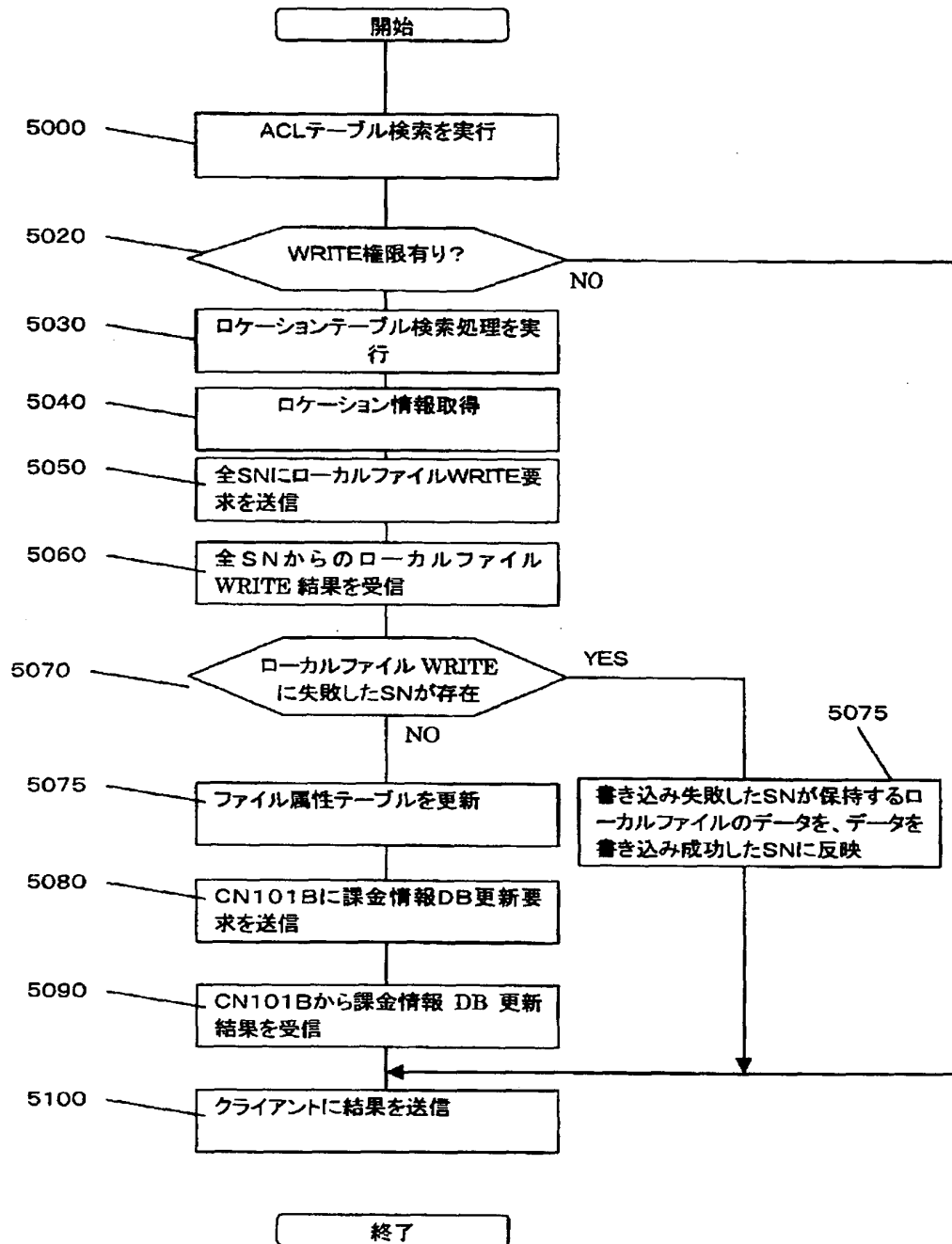
【図 6】

図 6



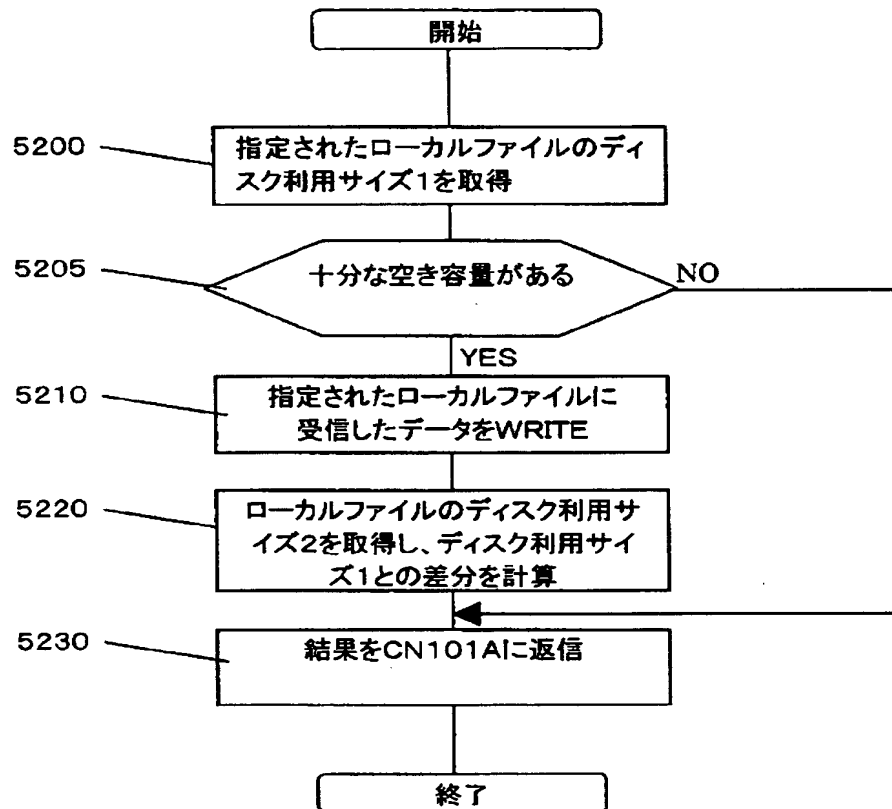
【図 7】

図 7



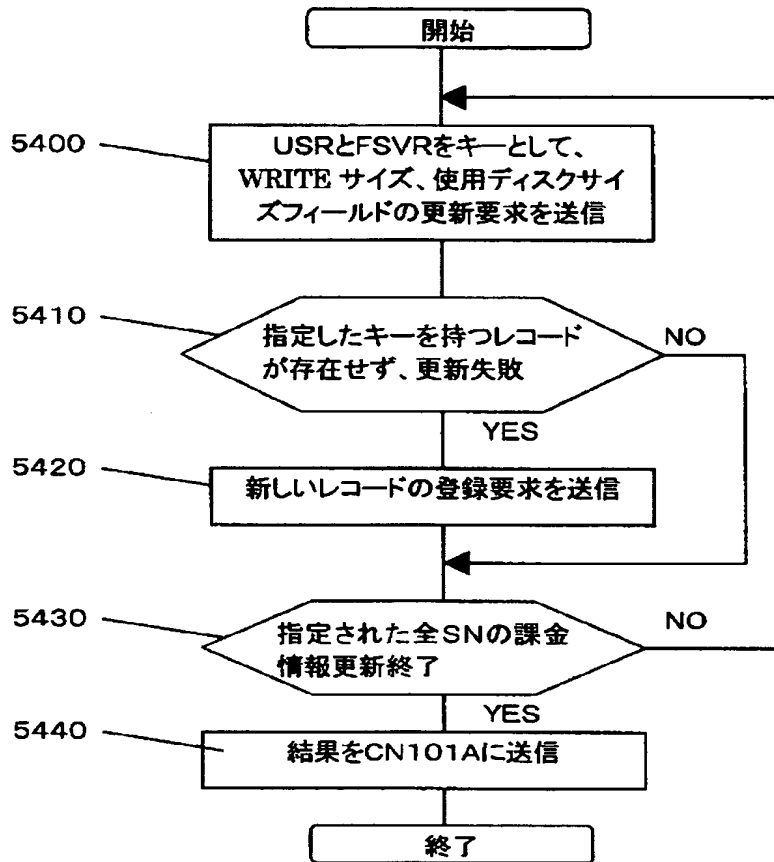
【図 8】

図 8



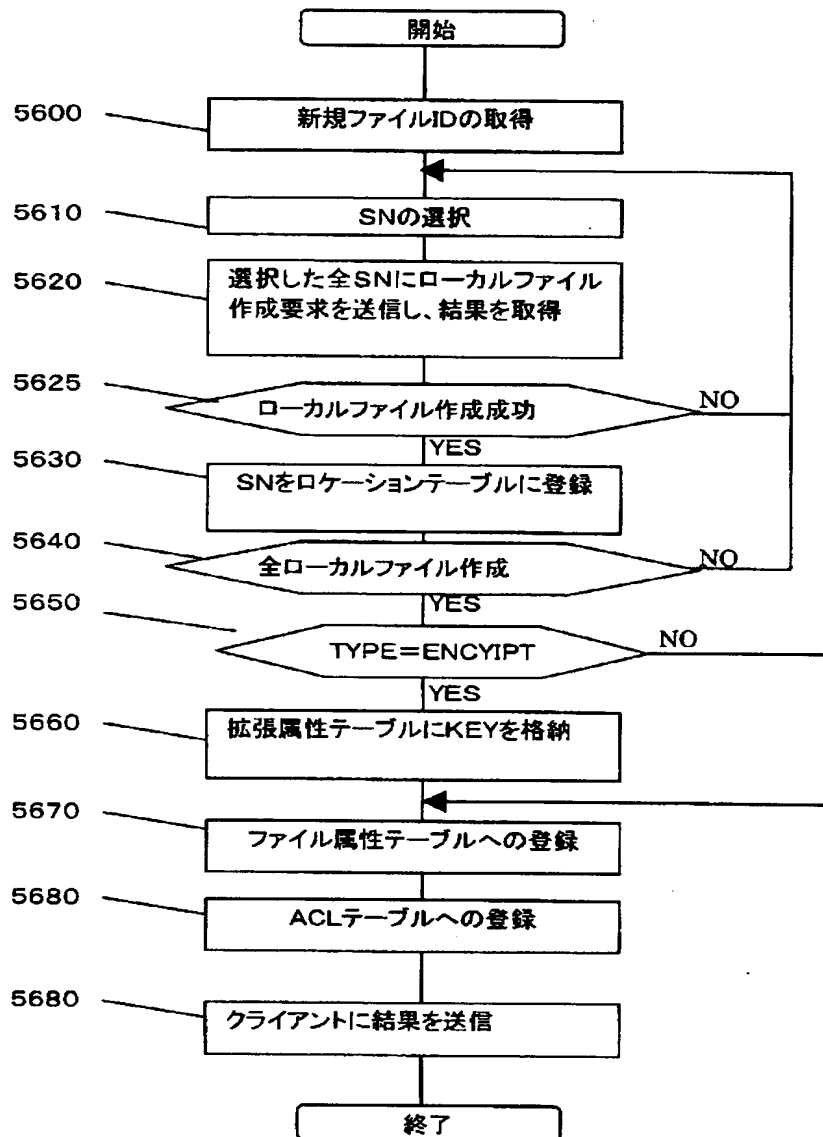
【図 9】

図9



【図10】

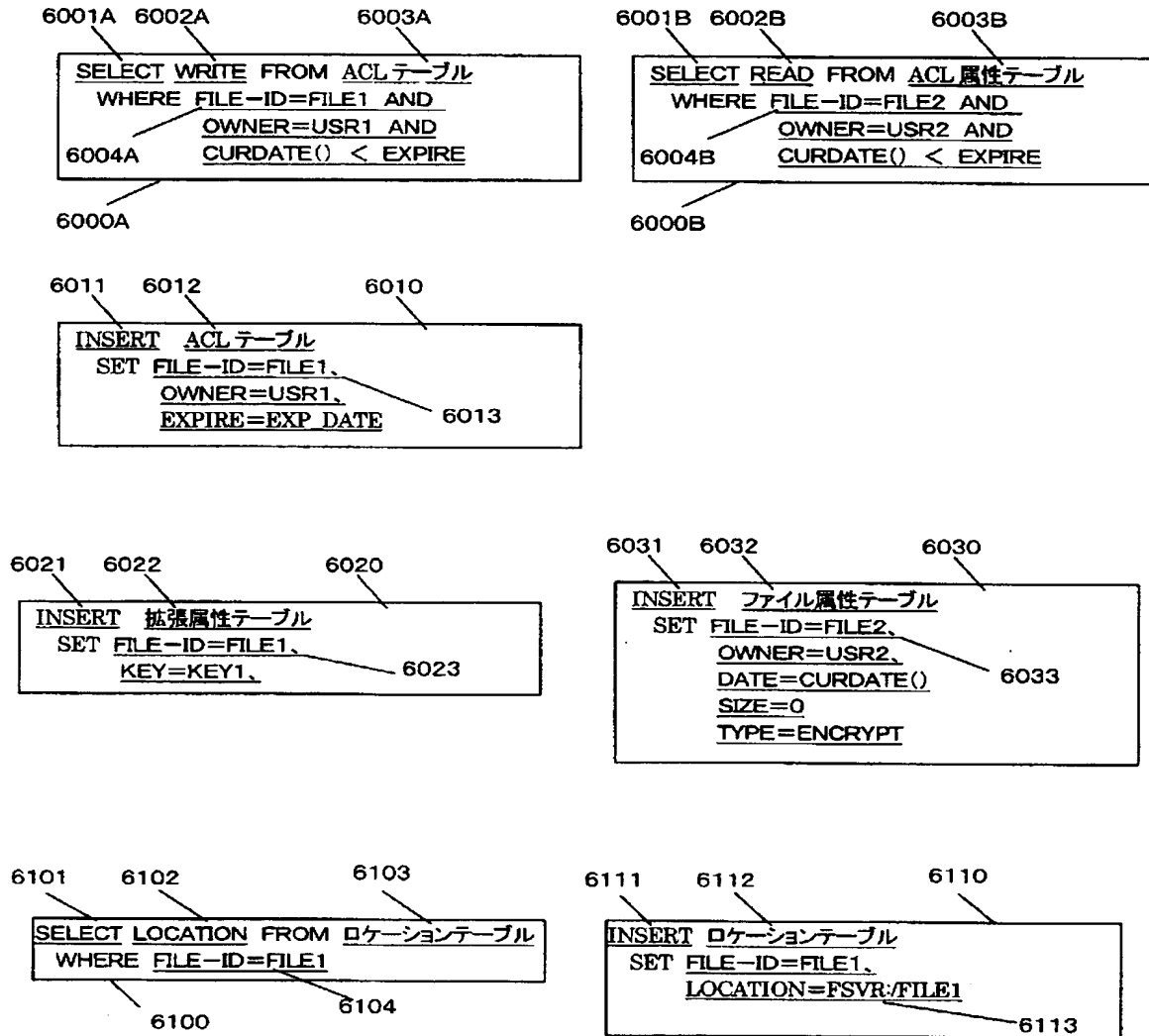
図10





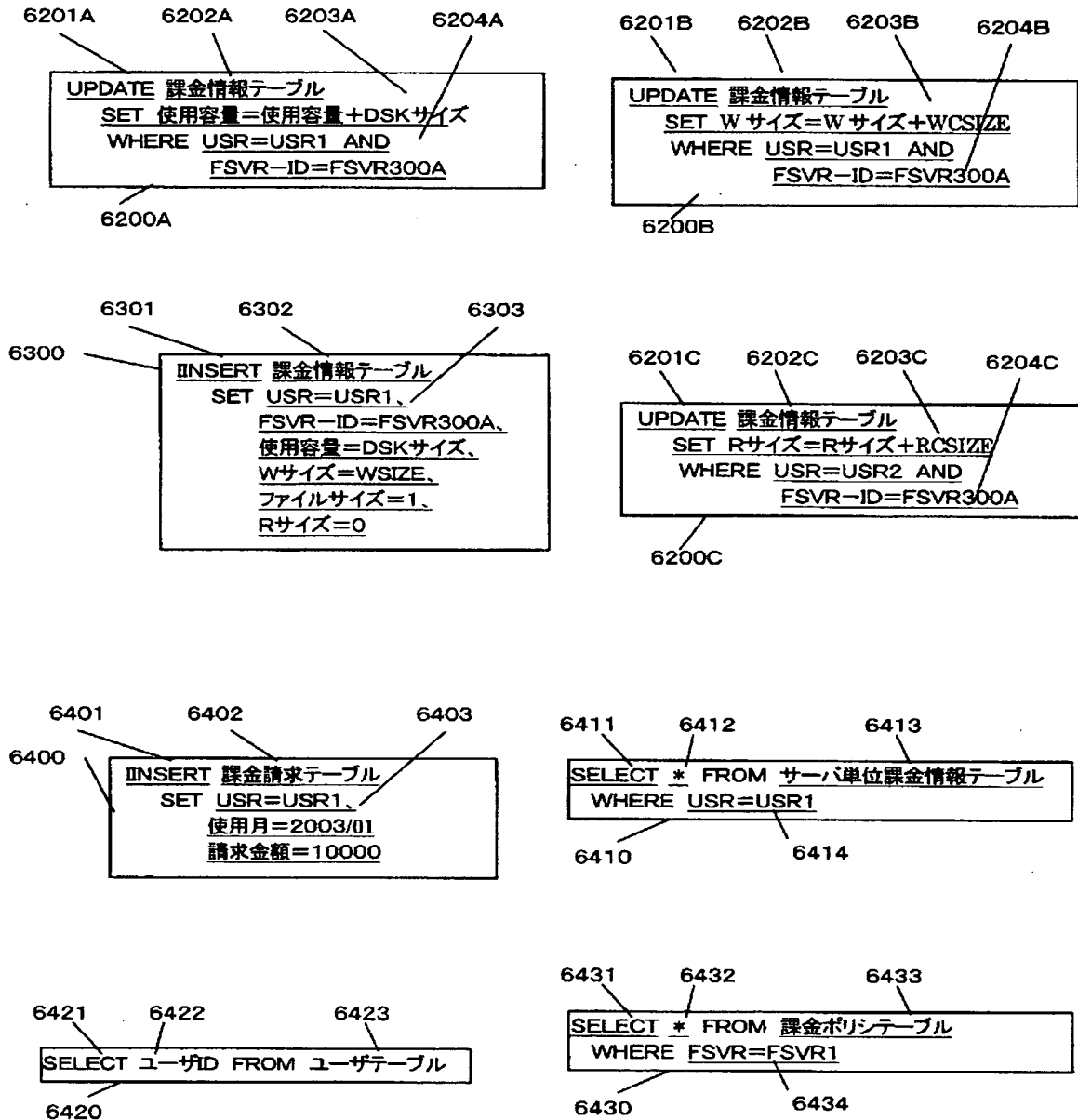
【図 11】

図 11



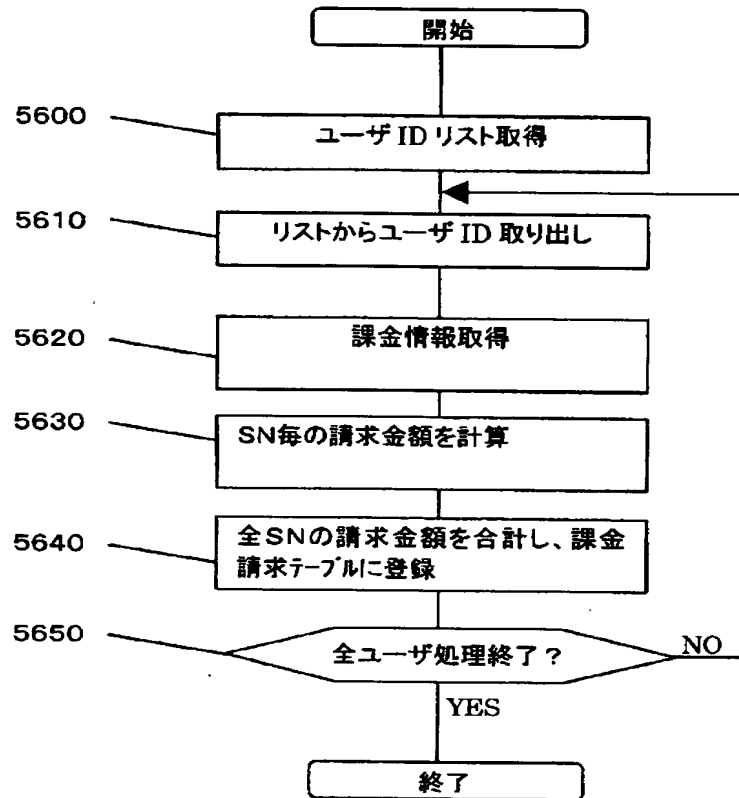
【図 12】

図 12



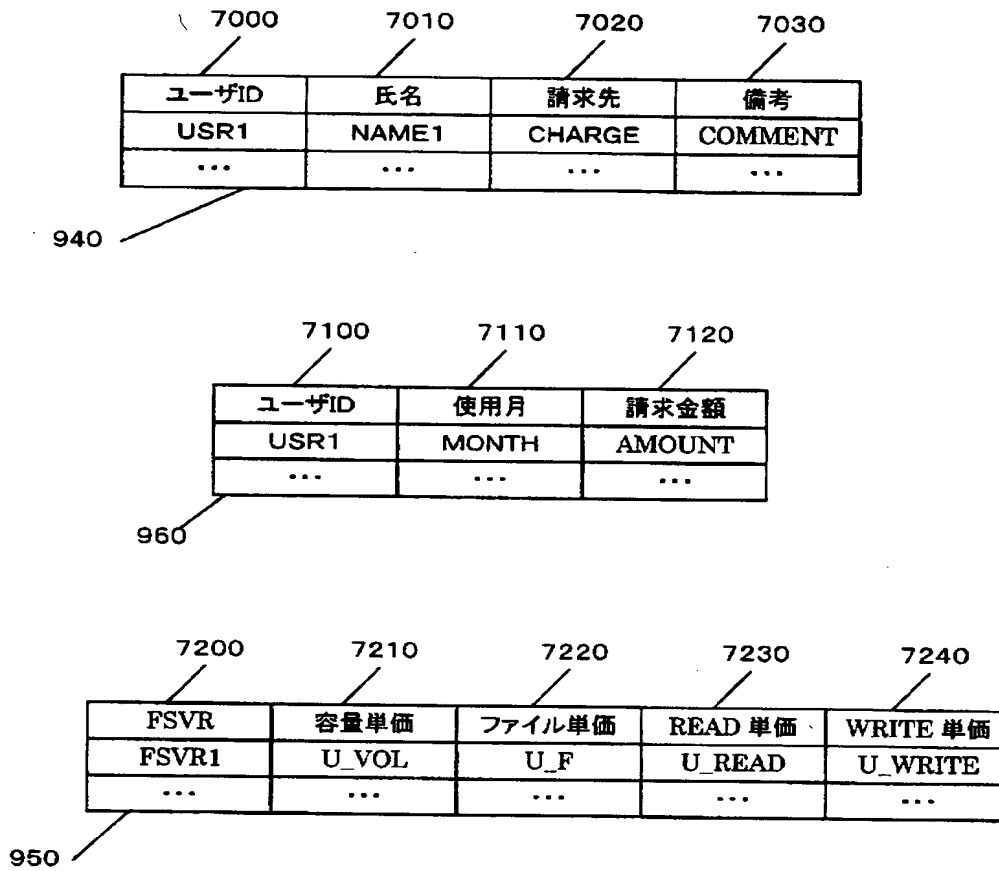
【図 13】

図 13



【図 14】

図 14



【書類名】 要約書

【要約】

【課題】 従来のファイルシステムでは、ひとつのファイルに対する多数ユーザのアクセス権限情報設定や、ファイル毎に異なるファイル属性情報の設定は、管理領域を浪費する問題があった。また、複数のストレージを用いてファイルを格納する場合に、ストレージ毎の課金情報管理ができなかった。

【解決手段】 システムは、ファイル属性を管理するデータベースのファイル属性DBと、課金情報を管理するデータベースの課金情報DBとファイルデータを格納するローカルファイルシステムを持つ。課金情報DBはユーザあるいはグループとサーバの組み合わせ毎にレコードを持ち、新しいユーザやサーバが追加される毎にレコードを追加する。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 8 6 9 0 6
受付番号	5 0 3 0 0 5 0 0 3 7 8
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 3 月 2 8 日

< 認定情報・付加情報 >

【提出日】	平成15年 3月27日
-------	-------------

次頁無

特願 2003-086906

出願人履歴情報

識別番号

[000005108]

1. 変更年月日

1990年 8月31日

[変更理由]

新規登録

住所

東京都千代田区神田駿河台4丁目6番地

氏名

株式会社日立製作所